Mobile Development
With Xcode for J2ME Devices

Session 627

These are confidential sessions—please refrain from filming or taking pictures.

Enterprise IT

Mobile Development With Xcode for J2ME Devices

Ted Jucevic
DTS Engineer

http://developer.apple.com/wwdc2005/

Mobile Development With Xcode for J2ME Devices

Michael Powers
Chief Technology Officer
mpowerplayer, Inc.

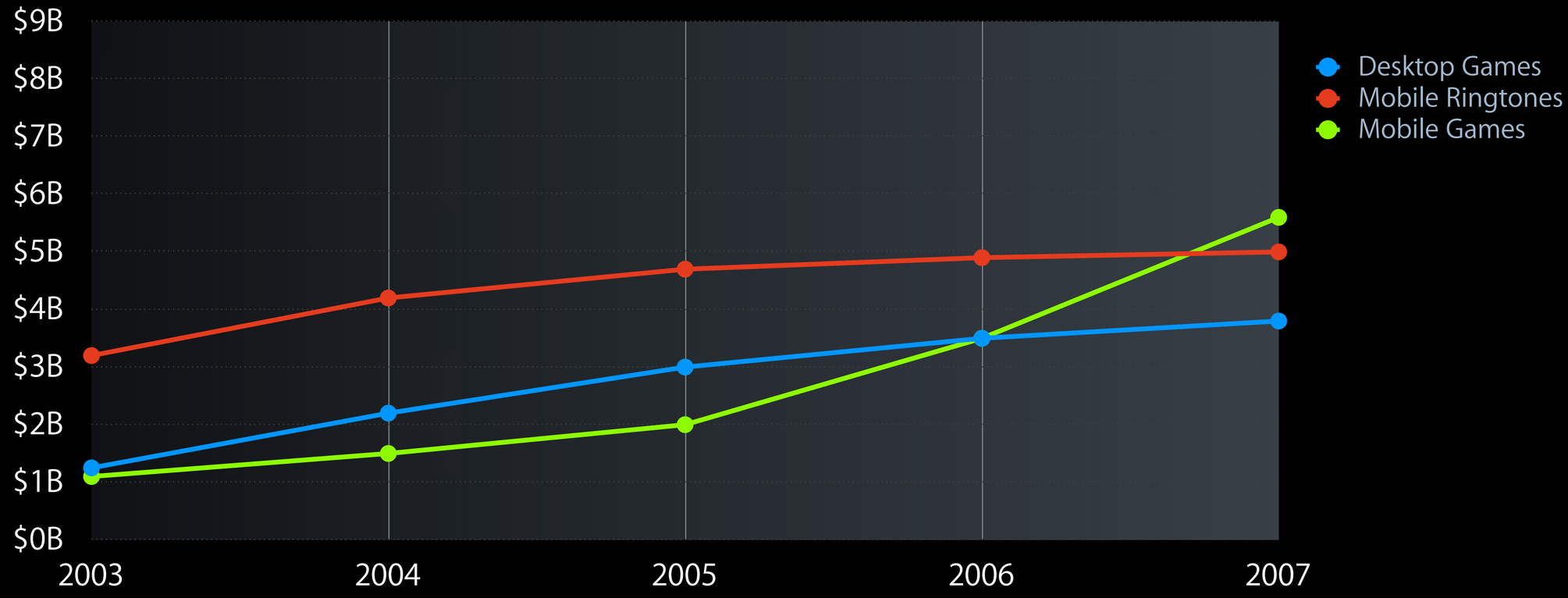
http://developer.apple.com/wwdc2005/

Mobile Development

What is it good for?

- Personal: the “third screen”
  - Keys, Wallet, Phone
  - Individual: not shared, personalized
  - Demographics: 50/50: m/f young/old
- Connected
  - Web services: think Dashboard Widgets
  - Thin Clients: think MathME
- Entertaining
  - “Casual games”: pick-up-and-put-down puzzlers
  - Moore’s law in full effect: playstation-quality 3D graphics
  - Games are a very hot market

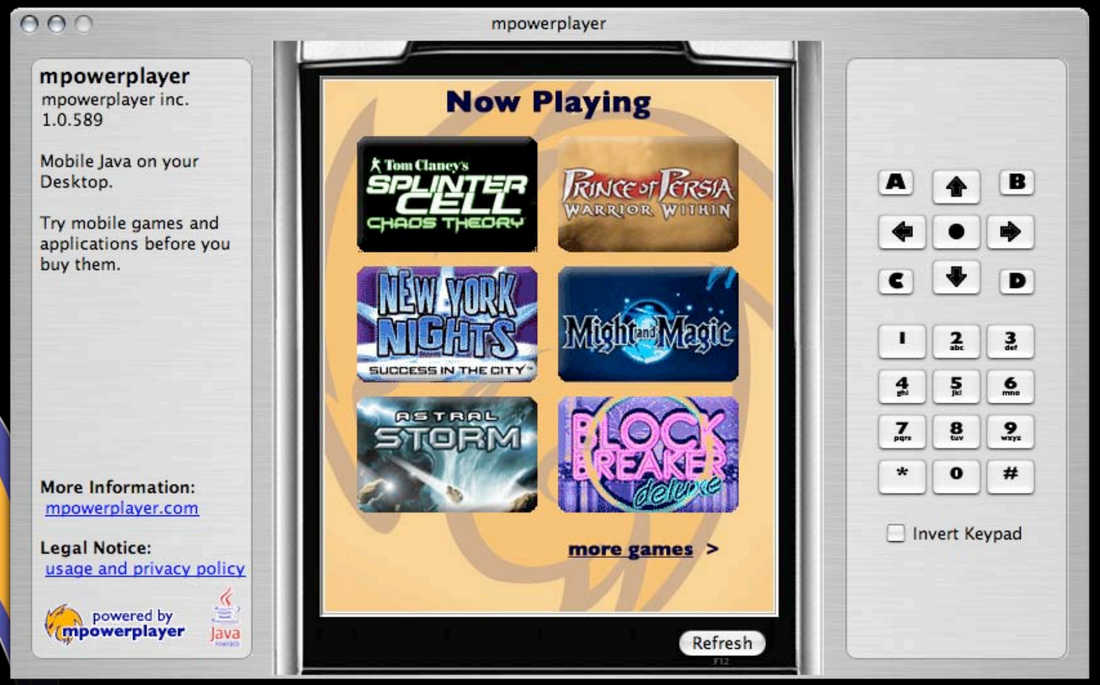
Market for Entertainment Software



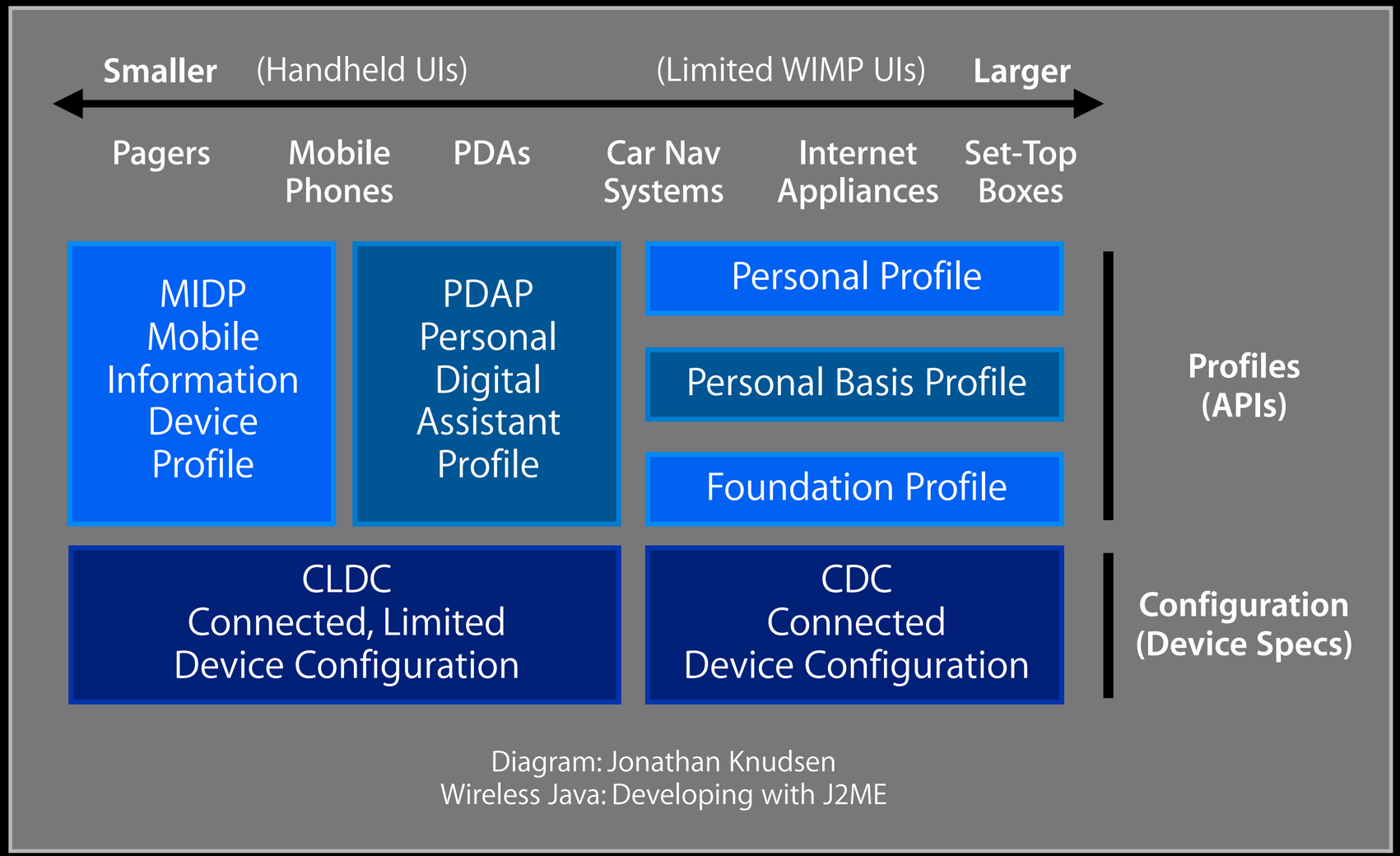
Source: PriceWaterhouseCoopers, IDC

Mobile Application Demos

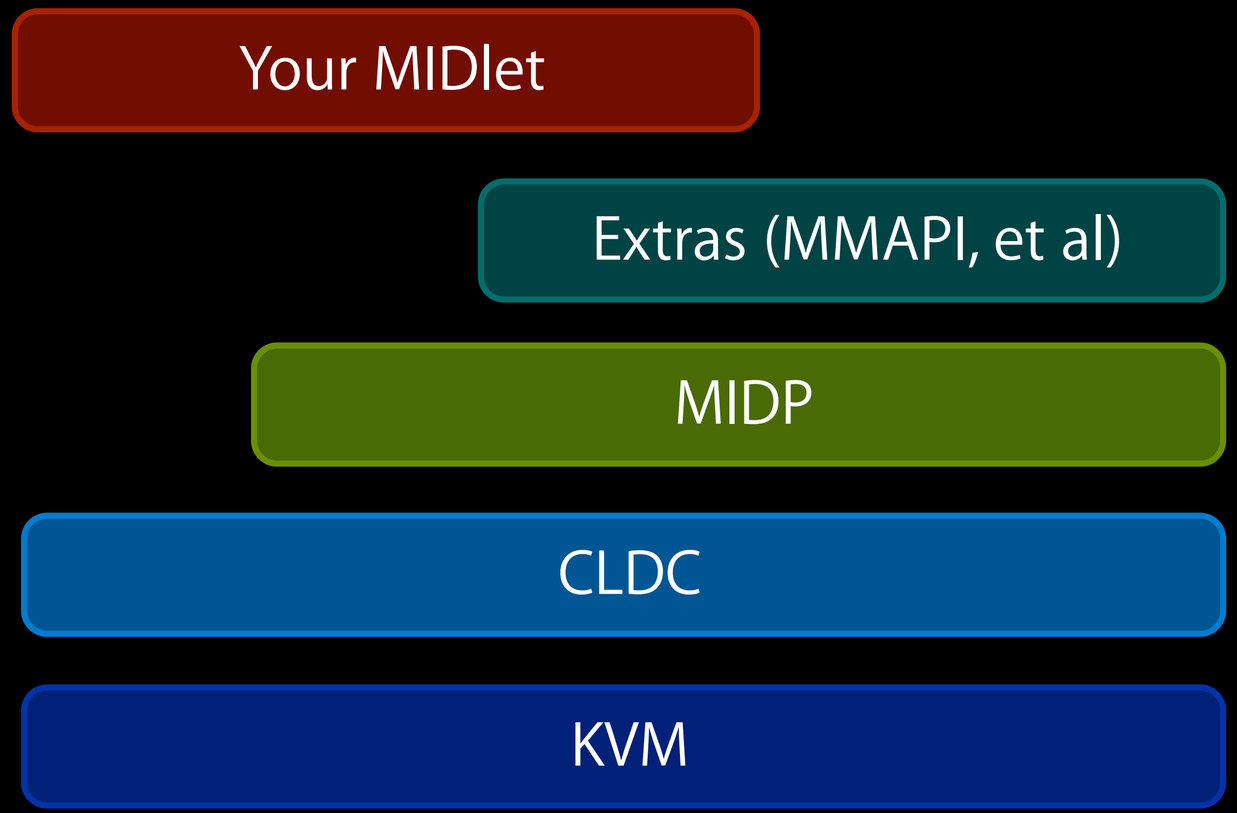
Powered by mpowerplayer



J2ME Universe



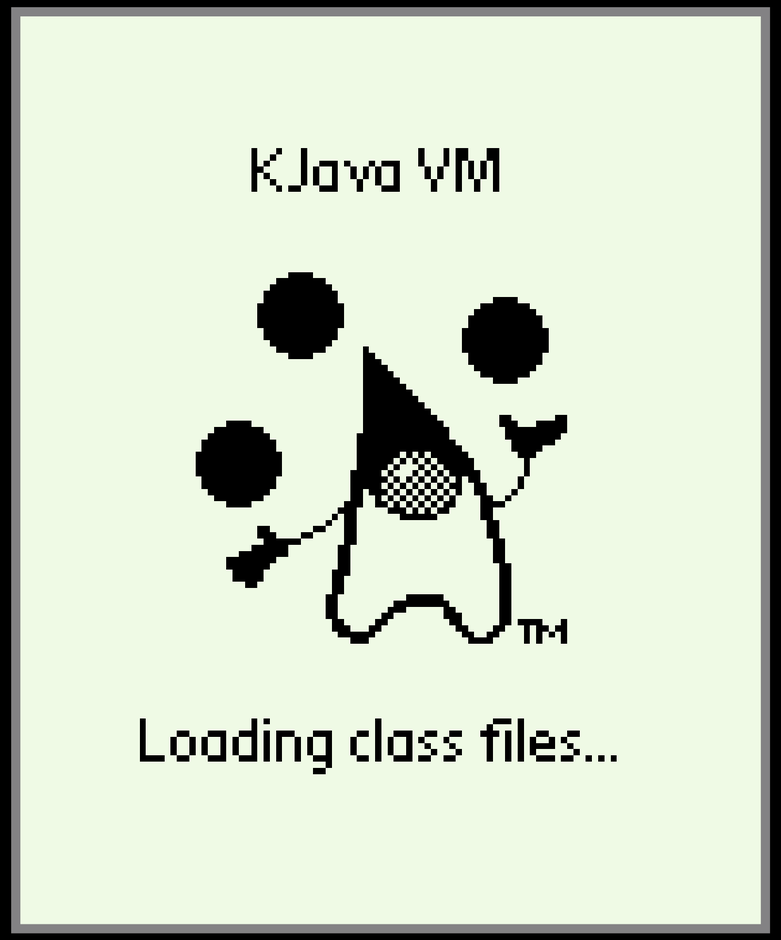
MIDlet Technology Stack



Virtual Machine (KVM)

The “k” virtual machine

- Minimum Specs
  - 32k heap, 8k persistent storage
  - 96x54 pixels (no color required)
  - http networking
- Typical Specs circa 2004
  - 200k heap, 5Mb persistent storage
  - 128x128 pixels, 4096 color
  - http, https, some have sockets
- 2005 and beyond!
  - Megs of heap, QVGA resolution, 3D hardware acceleration



Connected Limited Device Configuration (CLDC)

- Defines the required core features of the mobile runtime
- CLDC 1.0
  - Subset of the Java Language Specification
    - Portions of java.lang, java.io, java.util
    - Missing: Reflection, Serialization, File IO, Lists/Maps/Sets
- CLDC 1.1
  - Adds: Floating Point, Weak References
- Note: MIDP 2.0 does not require CLDC 1.1

MIDP 1.0

250 Million handsets in use today

- Low-level UI
  - Input: Handset Keypad and Pointer abstraction
  - Output: Canvas with basic graphics primitives
- High-level UI
  - Standard Widgets and Form API
- Networking: HTTP
- Persistence support: simple byte-array storage
- Image support: PNG and optional JPEG
- Audio support: none (!)

MIDP 2.0

Deployed and growing fast—One billion by 2007

- Low-level UI: new "Game API"
  - Sprites, Layers, region copy, mutable images, RGB access
- High-level UI
  - Custom UI Widgets, access to native themes
- Networking
  - HTTPS, Sockets, SSL/TLS
  - "Push" registry: launch in response to incoming connection
- Persistence: shared record stores
- Audio support: ABB and optional MMAPI (akin to JMF)
  - AIFF and WAV required, MIDI, AMR, MP3 optional

MIDP Optional Packages

- Mobile Media API (MMAPI)
  - Similar architecture to JMF: Players and Controls
  - Audio playback and capture, image capture, video playback
- Scalable 2D Graphics (M2G: JSR 226)
  - Playback and manipulation of SVG content
  - Support for rendering vector graphics in Graphics/Canvas
- Mobile 3D Graphics (M3G: JSR 184)
  - Scenegraph API similar to Java3D
  - Simple, cross-platform, 3D API suitable for games
- Supported by newer handsets from
  - Motorola, Nokia, SonyEricsson, others too

JSR-184 Demo

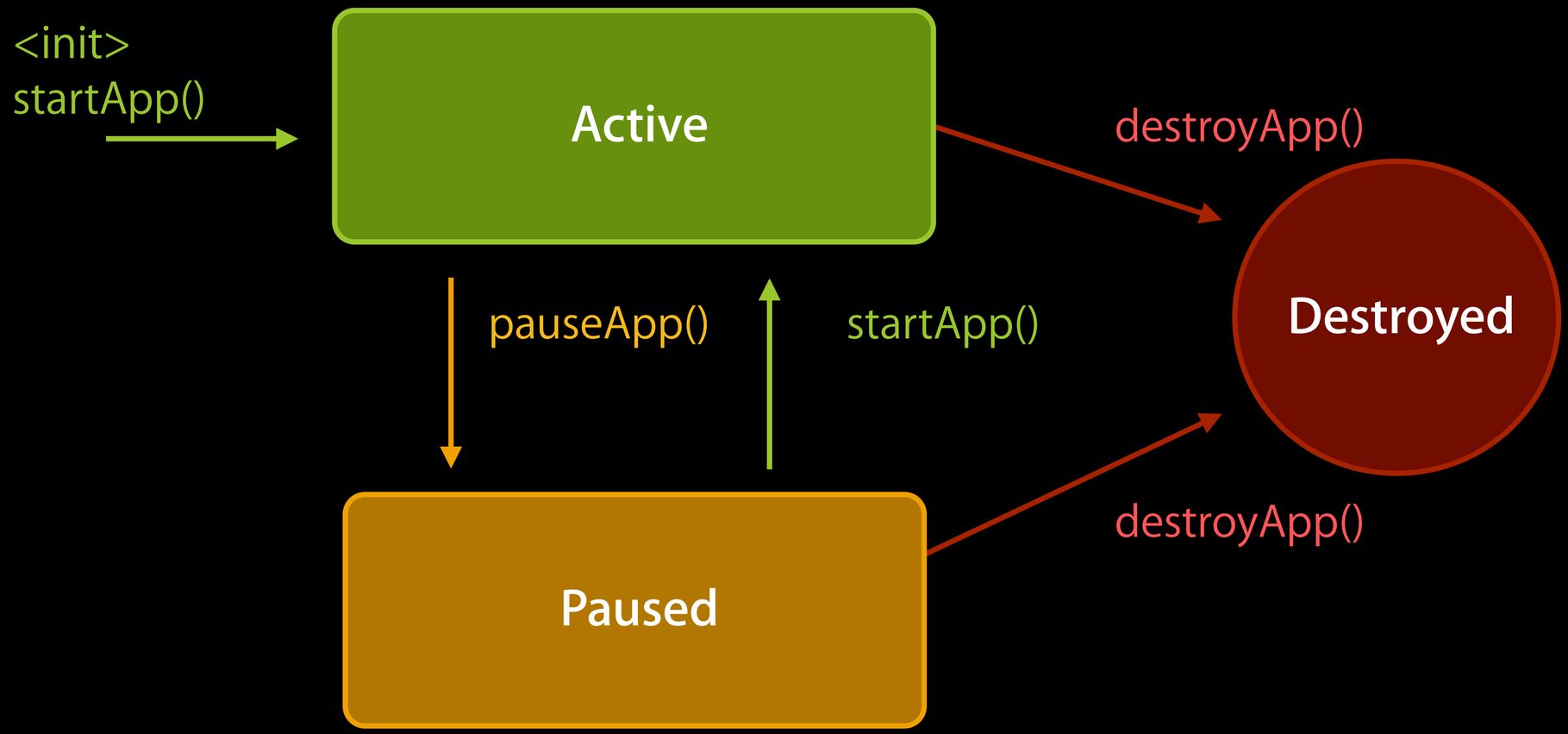
Mobile 3D Graphics for J2ME

What Is a MIDlet?

- JAR file
  - All classes, images, sounds, resources
  - Entry point: a subclass of javax.microedition.midlet.MIDlet
  - Classes must be “preverified”, may be signed
  - Typically 100s of K in size, but some devices enforce 64k limit

- JAD file: “Java Application Descriptor”
  - Text file in Properties format
  - Metadata for the application, duplication of data from Manifest
  - Name, Vendor, Version, Size, Requirements
  - A “preview” for devices to download before downloading JAR

MIDlet Life Cycle



MIDlet Build Cycle

Build	Preverify	Obfuscate	Jad: correct jar size	Run: mpowerplayer	Profile: Shark	Debug: Xcode	Deploy

Post-Build Targets

- Preverify
  - First step in J2ME Security
  - Annotates class files
  - Enables fast verification on mobile device
  - Required step in build process
- Obfuscate
  - Useful to reduce the deployed size of MIDlet
  - A number of options
    - Jode, ProGuard
    - Free, open source, and/or commercial
- JAD file
  - Automate the population of the MIDlet-Jar-Size property

Run With mpowerplayer

- Where it is
  - http://mpowerplayer.com/sdk
  - Free as in beer
- What it is
  - Pure Java J2ME Emulator/SDK
    - MIDP 2.0
    - MMAPI, M2G, M3G, Bluetooth
- What it isn't
  - Not the WTK: no packet sniffer, memory profiler, speed throttler
  - Not an IDE: we focus on what we do best
- Integrates with your existing tool chain

How It’s Done

Ted Jucevic
DTS Engineer

http://developer.apple.com/wwdc2005/

Debug With Xcode

- Custom executable
  - Launch Emulator passing in URL to Jar
- Debug Emulator process
  - Set breakpoints in your code
  - Debug Java like normal

Profile With Shark

- Setting the emulator up for profiling
  - -XrunShark
- Classes to exclude
  - java.awt.*,javax.swing.*, etc.
- Realities of profiling an emulator
- Packages help

Deploy to Device

- Bluetooth
  - JAR only
  - Mainly for testing
  - Bluetooth File Exchange App
    - /Applications/Utilities
    - Push over pre-verified Jar
- HTTP
  - Closer to real world
  - Requires Static IP
  - Works with Apache

Configuring Apache

- Add to mime.types file
  - text/vnd.sun.j2me.app-descriptor jad
  - application/java-archive jar
- Mac OS X Client
  - /etc/httpd/mime.types

Hands-on: What Are We Going to Do?

- Build a game using MIDP 2.0
- In the Emulator
  - Run the MIDlet
  - Debug the MIDlet
  - Shark the MIDlet
  - Deploy via Bluetooth

Develop, Debug, and Deploy

http://developer.apple.com/wwdc2005/

More Information

http://developer.apple.com/wwdc2005/

Documentation, Sample Code, and Other Resources

http://developer.apple.com/wwdc2005/

WWDC2005

Worldwide Developers Conference